Literatur- & Quellenverzeichnis

[AHDTPP] Hunt, Andrew; Thomas, David: *The Pragmatic Programmer : From journeyman to master.*
16th printing o.O.: Addiso-Wesley, 2005

[AWDWR] Thomas, Dave; Heinemeier Hanson, David: *Agile Web Development with Rails, 2nd Edition.*
„P2.01 printing“ o.O.: Pragmatic Programmers, 2007
(Digitale Version im Portable Document Format)

[CFRR] Fowler, Chad: *Rails Recipes.*
„P1.0 printing“ o.O.: Pragmatic Programmers, 2006
(Digitale Version im Portable Document Format)

[DBRFR] Black, David A.: *Ruby for Rails.*
Greenwich: Manning, 2006
(Digitale Version im Portable Document Format)

[DTPR] Thomas, Dave; Fowler, Chad; Hunt, Andy: *Programming Ruby : The Pragmatic Programmers' Guide.*
„9th printing“ o.O.: Pragmatic Programmers, 2006
(Digitale Version im Portable Document Format)

[HHDRD] Huxel, Harald: *Datenmigration relationaler Datenbanken.*
Darmstadt; Hochschule Darmstadt Fachbereich Informatik, Bachelorarbeit, 2007

[IEEE1219] Institute of Electrical and Electronics Engineers: *IEEE 1219 : IEEE Standard for Software Maintenance.*
Revised o.O.: IEEE Press, 1998

[ISO12207] International Organization for Standardization: *ISO/IEC 12207 : Standard for Information Technology-Software Life Cycle Processes.*
1995

[JKRP] Kerievsky, Joshua: *Refactoring to Patterns.*
4th printing o.O.: Addison-Wesley, 2006

[JWOSB] Weirich, Jim: *Symbols are not immutable strings.*
http://onestepback.org/index.cgi/Tech/Ruby/SymbolsAreNotImmutableStrings.red
27. Dezember 2005
zuletzt besucht am 12. Mai 2007

[MFPEAA] Fowler, Martin: *Pattern of Enterprise Application Architecture.*
8th printing o.O.: Addison-Wesley, 2005

[MFR] Fowler, Martin: *Refactoring : Improving the Design of Existing Code.*
19th printing o.O.: Addison-Wesley, 2006

[KINTT] Iverson, Ken E.: *Notation as a Tool of Thought.*
ACM Turing Award Lecture
delivered at ACM '79, Detroit

[RFREST] Fielding, Roy: *Architectural Styles and the Design of Network-based Software Architectures.*
Irvine: University of California, Dissertation, 2000
http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
zuletzt besucht am 12. Mai 2007

[SARV] Videodokumentation eines Treffens von Python- und Ruby-Programmieren: *Snakes and Rubies.*
Chicago, 3. Dezember 2005
http://www.djangoproject.com/snakesandrubies/
zuletzt besucht am 12. Mai 2007

[SEIS7] Sommerville, Ian: *Software Engineering, 7th Edition.* Version 2. Januar 2005
http://www.comp.lancs.co.uk/computing/resources/IanS/SE7/index.html
zuletzt besucht am 12. Mai 2007

[TORW2] O'Reilly, Tim: *What Is Web 2.0 : Design Patterns and Business Models for the Next Generation of Software.* 30. September 2005
http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
zuletzt besucht am 12. Mai 2007

[WKPD01] diverse Autoren: *Sapir-Whorf-Hypothesis.*
http://en.wikipedia.org/wiki/Sapir-Whorf_hypothesis
zuletzt besucht am 12. Mai 2007

[WKPD02] diverse Autoren: *Sapir-Whorf and programming languages.*
http://en.wikipedia.org/wiki/Sapir-Whorf_and_programming_languages
zuletzt besucht am 12. Mai 2007

[WKPD03] diverse Autoren: *Type system.*
http://en.wikipedia.org/wiki/Type_system
zuletzt besucht am 12. Mai 2007

[WKPD04] diverse Autoren: *Duck typing.*
http://de.wikipedia.org/wiki/Duck_Typing
zuletzt besucht am12. Mai 2007

[WKPD05] diverse Autoren: *Syntactic sugar.*
http://en.wikipedia.org/wiki/Syntactic_sugar
zuletzt besucht am 12. Mai 2007

[WL01] Holz, Patrick: *Was ist Web 2.0? : Designvorlagen und Geschäftsmodell für eine neue Software-Generation.*
http://twozero.uni-koeln.de/content/index_ger.html
zuletzt besucht am 12. Mai 2007

[WL02] Yakuel, Orli: *Sneak Peak at ajaxOS.*
28. April 2006
http://go2web2.blogspot.com/2006/04/sneak-peak-at-ajaxos.html
zuletzt besucht am 12. Mai 2007

[WL03] Gmail bzw. Google Mail:
http://gmail.com/
zuletzt besucht am 12. Mai 2007

[WL04] 37signals:
http://www.37signals.com/
zuletzt besucht am 12. Mai 2007

[WL05] Reporting Online:
http://www.reporting-online.com/
zuletzt besucht am 12. Mai 2007

[WL06] Cerezo, David: *On the Sapir-Whorf hypothesis and its relation to programming languages*.
9. Februar 2004
http://www.cerezo.name/archives/000005.html
zuletzt besucht am 12. Mai 2007

[WL07] Harfmann, Frank: *Zur Kritik der Sprachabhängigkeitsthese.*
http://www.netzgestalten.de/Frank.Hartmann/Sapir-Whorf.htm
zuletzt besucht am 12. Mai 2007

[WL08] Graphviz-Projekt:
http://www.graphviz.org/
zuletzt besucht am 12. Mai 2007

[WL09] Ruby-Projekt:
http://ruby-lang.org
zuletzt besucht am 12. Mai 2007

[WL10] JRuby-Projekt:
http://jruby.codehaus.org/
zuletzt besucht am 12. Mai 2007

[WL11] Builder-Projekt:
http://rubyforge.org/projects/builder/
zuletzt besucht am 12. Mai 2007

[WL12] Ruby-on-Rails-Projekt:
http://www.rubyonrails.com/
zuletzt besucht am 12. Mai 2007

[WL13] Ruby on Rails API:
http://api.rubyonrails.com/
zuletzt besucht am 12. Mai 2007

[WL14] Dumbill, Edd (O'Reilly Network): *Ruby on Rails : An Interview with David Heinemeier Hansson.*
30. August 2005
http://www.oreillynet.com/pub/a/network/2005/08/30/ruby-rails-david-heinemeier-hansson.html
zuletzt besucht am 12. Mai 2007

[WL15] Hibernate-Projekt:
http://hibernate.org/
zuletzt besucht am 12. Mai 2007

[WL16] Heinemeier Hansson, David: *Choose a single layer of cleverness.*
27. September 2005
http://www.loudthinking.com/arc/000516.html
zuletzt besucht am 15. April 2007

[WL17] Fowler, Martin: *DatabaseStyles.*
24. Mai 2004
http://martinfowler.com/bliki/DatabaseStyles.html
zuletzt besucht am 12. Mai 2007

[WL18] RSpec-Projekt:
http://rspec.rubyforge.org/
zuletzt besucht am 12. Mai 2007

[WL19] Ambler, Scott W.: *Introduction to Test Driven Design.*
31. März 2007
http://www.agiledata.org/essays/tdd.html
zuletzt besucht am 12. Mai 2007

[WL20] Behaviour Driven Development:
http://behaviour-driven.org/
zuletzt besucht am 12. Mai 2007

[WL21] Agile Manifesto:
http://agilemanifesto.org/
zueletzt besucht am 12. Mai 2007

[WL22] Extreme Programming:
http://www.extremeprogramming.org/
zuletzt besucht am 12. Mai 2007

[WL23] Mongrel:
http://mongrel.rubyforge.org/
zuletzt besucht am 12. Mai 2007

[WL24] Capistrano (früher „Switchtower“)
http://www.capify.org/
zuletzt besucht am 12. Mai 2007

Darüber hinaus haben folgende Bücher die Arbeit geprägt:

Brooks, Frederick P. Jr.: *The Mythical Man-Month : Essays on Software Engineering.*
Anniversary Edition, 29th printing o.O.: Addison Wesley, 2006

Graham, Paul: Hackers & Painters : *Big Ideas from the Computer Age.*
Sebastol: O’Reilly, 2004

Quelltexte

#{RAILS_ROOT}/app/models/author.rb

```
class Author < ActiveRecord::Base
  belongs_to :bachelor_thesis
  validates_presence_of :first_name, :last_name
end
```

#{RAILS_ROOT}/app/models/bachelor_thesis.rb

```
class BachelorThesis < ActiveRecord::Base
  VALID_GRADES = ['1.0', '1.3', '1.7',
                  '2.0', '2.3', '2.7',
                  '3.0', '3.3', '3.7',
                  '4.0', '5.0', ''].freeze

  has_one  :author
  has_many :pages
  has_many :readings
  has_many :readers, :through => :readings

  validates_inclusion_of  :grade, :in => VALID_GRADES
  validates_presence_of   :title, :due_date
  validates_uniqueness_of :title

  def before_validations_of_create
    self.due_date ||= 3.months.from_now
  end
end
```

#{RAILS_ROOT}/app/models/page.rb

```
class Page < ActiveRecord::Base
  belongs_to :bachelor_thesis
  acts_as_list :scope => :bachelor_thesis

  validates_presence_of :content
  validates_presence_of :bachelor_thesis_id
end
```

Quelltexte (Fortsetzung)

#{RAILS_ROOT}/app/models/reader.rb

```
class Reader < ActiveRecord::Base
  has_many :readings

  has_many :bachelor_theses, :through => :readings

  has_many :unfinished_theses,
           :through => :readings,
           :source => :bachelor_thesis,
           :conditions => { :finished_reading => false }

  has_many :unfinished_theses,
           :through => :readings,
           :source => :bachelor_thesis,
           :conditions => { :finished_reading => false }

  validates_presence_of :first_name, :last_name
end
```

#{RAILS_ROOT}/app/models/reading.rb

```
class Reading < ActiveRecord::Base
  belongs_to :reader
  belongs_to :bachelor_thesis

  validates_presence_of :finished_reading
end
```

Quelltexte (Fortsetzung)

#{RAILS_ROOT}/db/migrate/001_add_sessions.rb
(in dieser Arbeit nicht behandelt)

```
class AddSessions < ActiveRecord::Migration
  def self.up
    create_table :sessions do |t|
      t.column :session_id, :string
      t.column :data, :text
      t.column :updated_at, :datetime
    end

    add_index :sessions, :session_id
    add_index :sessions, :updated_at
  end

  def self.down
    drop_table :sessions
  end
end
```

#{RAILS_ROOT}/app/models/002_create_bachelor_theses.rb

```
class CreateBachelorTheses < ActiveRecord::Migration
  def self.up
    create_table :bachelor_theses do |t|
      t.column :title, :string
      t.column :subtitle, :string
      t.column :author_name, :string
      t.column :grade, :string
      t.column :due_date, :datetime
    end
  end

  def self.down
    drop_table :bachelor_theses
  end
end
```

Quelltexte (Fortsetzung)

#{RAILS_ROOT}/db/migrate/003_create_authors.rb

```
class CreateAuthors < ActiveRecord::Migration
  def self.up
    create_table :authors do |t|
      t.column 'first_name', :string
      t.column 'last_name', :string
      t.column 'bachelor_thesis_id', :integer
    end

    theses = BachelorThesis.find(:all)

    theses.each do |thesis|
      first_name, last_name = thesis.author_name.split(' ')
      new_author = Author.create :first_name => first_name,
                                 :last_name => last_name
      thesis.author = new_author
    end

    remove_column :bachelor_theses, :author_name
  end

  def self.down
    add_column :bachelor_theses, 'author_name', :string

    BachelorThesis.find(:all).each do |thesis|
      thesis.author_name = thesis.author.first_name + ' ' +
                           thesis.author.last_name
      thesis.save
    end

    drop_table :authors
  end
end
```

Quelltexte (Fortsetzung)

#{RAILS_ROOT}/db/migrate/004_create_pages.rb

```
class CreatePages < ActiveRecord::Migration
  def self.up
    create_table :pages do |t|
      t.column 'content', :text
      t.column 'bachelor_thesis_id', :integer
      t.column 'position', :integer
    end
  end

  def self.down
    drop_table :pages
  end
end
```

#{RAILS_ROOT}/app/models/005_create_readers.rb

```
class CreateReaders < ActiveRecord::Migration
  def self.up
    create_table :readers do |t|
      t.column :first_name, :string
      t.column :last_name, :string
    end

    create_table :readings do |t|
      t.column :bachelor_thesis_id, :integer
      t.column :reader_id, :integer
      t.column :finished_reading, :boolean, :default => false
    end
  end

  def self.down
    drop_table :readings
    drop_table :readers
  end
end
```

Quelltexte (Fortsetzung)

RSpec-Beispiel (aus Kapitel 5; RSpec-Version 0.9):

```
require File.dirname(__FILE__) + '/spec_helper'
require File.dirname(__FILE__) + "/stack"

describe "non-empty Stack", :shared => true do
  it { @stack.should_not be_empty }

  it "should return the top item when sent #peek" do
    @stack.peek.should == @last_item_added
  end

  it "should NOT remove the top item when sent #peek" do
    @stack.peek.should == @last_item_added
    @stack.peek.should == @last_item_added
  end

  it "should return the top item when sent #pop" do
    @stack.pop.should == @last_item_added
  end

  it "should remove the top item when sent #pop" do
    @stack.pop.should == @last_item_added
    unless @stack.empty?
      @stack.pop.should_not == @last_item_added
    end
  end
end

describe "non-full Stack", :shared => true do
  it { @stack.should_not be_full }

  it "should add to the top when sent #push" do
    @stack.push "newly added top item"
    @stack.peek.should == "newly added top item"
  end
end

describe Stack, " (empty)" do
  before(:each) do
    @stack = Stack.new
  end

  it { @stack.should be_empty }

  it_should_behave_like "non-full Stack"

  it "should complain when sent #peek" do
    lambda { @stack.peek }.should raise_error(StackUnderflowError)
  end

  it "should complain when sent #pop" do
    lambda { @stack.pop }.should raise_error(StackUnderflowError)
  end
end
```

Quelltexte (Fortsetzung)

RSpec-Beispiel (Fortsetzung):

```
describe Stack, " (with one item)" do
  before(:each) do
    @stack = Stack.new
    @stack.push 3
    @last_item_added = 3
  end

  it_should_behave_like "non-empty Stack"
  it_should_behave_like "non-full Stack"
end

describe Stack, " (with one item less than capacity)" do
  before(:each) do
    @stack = Stack.new
    (1..9).each { |i| @stack.push i }
    @last_item_added = 9
  end

  it_should_behave_like "non-empty Stack"
  it_should_behave_like "non-full Stack"
end

describe Stack, " (full)" do
  before(:each) do
    @stack = Stack.new
    (1..10).each { |i| @stack.push i }
    @last_item_added = 10
  end

  it { @stack.should be_full }

  it_should_behave_like "non-empty Stack"

  it "should complain on #push" do
    lambda {
      @stack.push Object.new
    }.should raise_error(StackOverflowError)
  end
end
```

Datenbank-Dokumentation

Folgendes SQL-Skript beschreibt die Datenbank-Struktur, wie sie mit den gezeigten Migrations von Ruby on Rails in einer MySQL-Datenbank (Version 5.0.23) angelegt wird.

```
CREATE TABLE `bachelor_theses` (
  `id` int(11) NOT NULL auto_increment,
  `title` varchar(255) default NULL,
  `subtitle` varchar(255) default NULL,
  `grade` varchar(255) default NULL,
  `due_date` datetime default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `authors` (
  `id` int(11) NOT NULL auto_increment,
  `first_name` varchar(255) default NULL,
  `last_name` varchar(255) default NULL,
  `bachelor_thesis_id` int(11) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `pages` (
  `id` int(11) NOT NULL auto_increment,
  `content` text,
  `bachelor_thesis_id` int(11) default NULL,
  `position` int(11) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `readers` (
  `id` int(11) NOT NULL auto_increment,
  `first_name` varchar(255) default NULL,
  `last_name` varchar(255) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `readings` (
  `id` int(11) NOT NULL auto_increment,
  `bachelor_thesis_id` int(11) default NULL,
  `reader_id` int(11) default NULL,
  `finished_reading` tinyint(1) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Kolophon

Diese Bachelorarbeit ist mit dem freien DTP-Programm „Scribus“ (zu finden unter http://www.scribus.net/) in den Versionen 1.3.3.7(sic!), 1.3.3.8 und 1.3.3.9 erstellt und gesetzt. Der Export nach PDF erfolgte mit den programmeigenen Routinen.

Abgesehen von den Graphviz-Diagrammen aus Kapitel 1 und den Tabellen wurden alle Abbildungen in Microsoft Visio 2007 erzeugt.

Als Schriftart wird die Lucida-Familie verwendet: Der normale Text ist in „Lucida Fax“ gehalten (11pt mit 16pt Zeilenabstand), während die Anmerkungen und Fussnoten in der Schriftart „Lucida Sans“ gesetzt wurden. Die Quelltexte (9pt/10pt) werden in „Lucida Console“ dargestellt; die Zitate verwenden „Lucida Calligraphy“.